ERROR HANDLING AND LOGGING AND DEBUGGING

1. Errors can be handled through an __init__ method: assessing the data requirements before they are initialised, and they can also be handled through the @property decorator using setter, getter and delete. The builtin Exception can be used to create custom errors. Descriptors can also be used.

2. Logging is handled through Logging module, but fisrt use print statements at appropriate places and replace them with logging module. Even decorators can be very useful in creating logs

3. Debug Python use pdb, and C code use gdb & readelf for C code.

- Start from the beginning to establish program entry.
- Always list the surrounding source code
- If possible get an overview of the process address space – including mapped modules, stack and heap
- preferrably step through code and follow the debugger when it calls functions on the stack frame and listing the surrounding source code to observe its pattern
- Print available variables contents and address.
- Determine stack frames addresses and all available running threads.
- When stepping through the code observe the stack frames when a new function is called and print contents of memory addresses.